Technology Day Event

Imperva, Inc. presents: Web Application and Database Security

March 28, 2005

IMPERVA

Goal for Today

Technology Day Focus:

1. Raise Awareness - around the threats that target your critical applications and databases
2. Define the Technology Direction - in the IT Enterprise security market with a look at the history and future requirements

Agenda – March 28th, 2005

9:00 – 9:30	Web Application Security Attack Demonstration **Terry Ray** – Imperva, Sr. Security Engineer
9:30 – 10:15	Database Threats and Mitigation Landscape **Mark Kraynak** – Imperva, Dir. Product Marketing
10:15 – 10:30	Break
10:30 – 11:15	The Future of the Firewall **Shlomo Kramer** – Imperva CEO and Co-Founder of Check Point Software
11:15 – 11:30	Questions / Answers

Database Security: Threat Classification and Mitigation

Mark Kraynak, Director of Product Marketing, Imperva, Inc.

Teale Data Center Technology Day

March 28, 2005

iMPERVA

Agenda

- Database Security Today
- Pitfalls of Current Database Protection Approaches
- Database Security Threats Revisited
- Client Based Threat Classification Scheme
- Building an Effective Solution

Database Security Today – A Brief Overview

- Hardening of OS / DB platform including vendor-specific issues
  - Requires a system administrator
- Careful configuration of granular options within database server
  - Requires a database administrator
- Proper coding of applications
  - Requires the good will and attention of programmers
- Granular object and system privileges management within the database
  - Requires a database administrator working in tandem with the application programmer

Current Protection Approaches

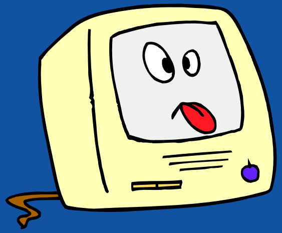
- The worst “worst case” assumption
  - Omnipotent Attacker (anyone can try anything)
  - Omnipresent Attacker (attacker can be anywhere)
  - Attack Pervasiveness (every possible attack will happen)
- Result: Protect Anything Using Any Available Technique
  - Attacker / defender asymmetric effort problem:
    An attacker need only find a single vulnerability while a defender must counter all possible threats.
- Consequences:
  - Dispersion of resources
  - Over budgeting
  - All attackers get the same (very limited) attention

Pitfalls of Current Protection Approaches

- Organizational Integration Required

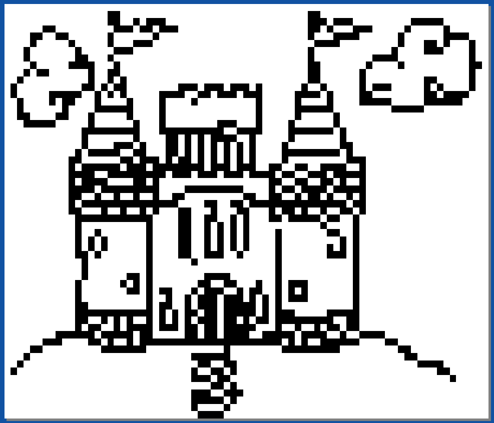
- Security Drives Capacity Planning

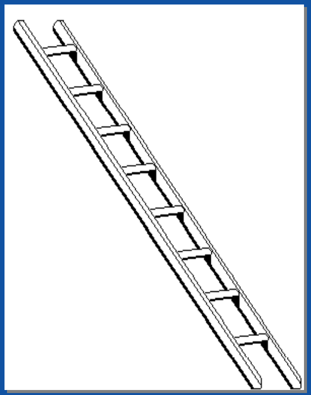
- Contradicting Requirements

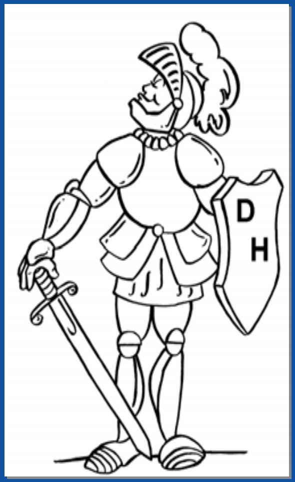


- Technical Shortcomings

Database Security Revisited







How many guards are required for our castle?

Protection Examples

- Given a database server connected to a web server
- Example - The worst “worst case” scenario
  - “Let’s protect the web application as though the database server is unprotected and protect the database as though the application is completely insecure”
  - Never enough resources
- Counter Example – Leveraging a layered approach
  - An attacker from the Internet cannot (and will not) perform login brute-forcing given that the perimeter FW is properly configured
  - Protection resources can be proportional

Client Based Threat Classification

Premise:

By analyzing the threat characteristics of each access type (client), we can achieve better results.

- Focus resources on actual threats
- Design a more effective protection scheme
- Simplify security management

Client Based Threat Classification

Assumptions

- Attackers will use simple methods before resorting to complex ones
- Attackers will use a technique only if it provides additional access
- Attackers will encounter unrelated security mechanisms before they can attack the database

Client Based Threat Classification

What are the Client Groups?

1. Internet Users
2. Intranet Users
3. Thick Client Users
4. Non-Users
5. Administrative Users

Internet Attack Example

1 - Online Retailer

- Company
  - Large online retailer
- Situation
  - “Hidden” error messages to avoid SQL Injection
    - Standard, generic error page meant to prevent reconnaissance
  - Attacker sends “wait” command to DB
    - Attacker can understand if the attempt worked by observation
- Solution
  - “Wait” command not in known set of queries
  - Block and log user actions

Client Based Threat Classification

1 - Internet Users

- Most Probable Attacks
  - SQL Injection
- Possible Attacks
  - Compromise Web server and launch network level attacks
  - Compromise Web server and launch an SQL client using the application credentials
- Unlikely Attacks
  - Compromise Web server and launch brute force login attacks on database

- Restrict protocols between app and DB
- Dedicated account for each application
- Restrict SQL access of app to known queries
- Zero-tolerance for failed logins
- Secure Web server platform
- IDS / IPS inspection

Intranet Attack Example

2 - Large Financial Institution

- Company:
  - Large International Bank
- Situation
  - Ex-employee attempted SQL injection via online banking
    - Attacked part of the application he had written himself
- Solution
  - Attempted query not in known query set
  - Block and log

Client Based Threat Classification

2 - Intranet Users

- Most Probable Attacks
  - SQL Injection
  - Privilege Abuse
- Possible
  - Compromise Web server and launch network level attacks
  - Direct network level attacks
  - Login brute force attack against the database
- Unlikely
  - Compromise web server and launch brute force login attacks on database
  - Complex network based attacks

- Restrict protocols between app and DB
- Dedicated account for each application
- Restrict SQL access of app to known queries
- Zero-tolerance for failed logins
- "Time of Day" Restrictions*
- Secure app server platform
- IDS / IPS Inspection
- Audit

Thick Client Attack Example

3 - eCommerce Company

- Company
  - Retailer of prepaid services with accounts managed online
- Situation
  - Financial loss due to white collar theft
    - Direct database attack by internal application developer
    - Credited accounts of associates with free services
- Solution
  - Granular audit of updates executed by developers (INCLUDING query parameters!)
  - Find relevant events

Client Based Attack Classification

3 - Thick Client Users

- Most Probable Attacks
  - Privilege Abuse (e.g. Copy all medical records to a memory stick)
  - Use alternative client software (e.g. SQL*Plus)
  - Generate arbitrary database access statements (DDL and DML) using the application’s credentials against non-application tables or against sensitive information
  - Connect to database using default accounts
- Other Possible Attacks
  - Brute-force credentials for privilege elevation
  - Launch network level attacks against database communication protocol
  - Launch network level attacks expecting internal network access controls to be loose
- Improbable Attacks
  - Launch complex network level attacks against database server

Client Based Attack Classification

3 - Thick Client Users (cont.)

- Restrict protocol access between clients and DB
- Dedicated account for each app role or client software
- Restrict use of the app account to the given client software*
- Restrict access of the app account(s) to known queries
- Restrict concurrent connections from each source IP
- Zero-tolerance for failed logins*
  - Depends on app structure
  - Pay special attention to default accounts
- “Time of Day” restrictions*
- IDS / IPS inspection
- Audit

Non user attack example

4 - Blaster Worm

- Situation
  - Blaster infections on server segments over (non-DB) protocols (Microsoft RPC)
- Solution
  - Deny non-DB protocols on DB segments
  - IPS / IDS on DB protocols

Client Based Threat Classification

4 - Non Users

- Most Probable Attacks
  - Use of default accounts or brute forcing database credentials
  - Simple network attacks as part of a non-targeted attack
  - Simple and complex network attacks targeted on the database server and probably the database protocol

- Restrict protocol access to database servers
- IDS / IPS inspection
- Detect login brute forcing
- Detect use of (otherwise unused) default accounts
- Detect the use of (otherwise unused) client software

Administrator Attack Example

5 - Online Services Company

- Company
  - Application Service Provider (service oriented applications)
- Situation
  - Admin user dropping tables during work hours
    - (not an attack, but against company policy)
- Solution
  - Granular auditing of admin user action
  - Gentle reminder of policy

Client Based Threat Classification

5 - Administrative Users

- Admin users are a “more special” case than others
- The aim is to positively identify administrative users’ actions rather than trying to stop them
- Administrative users include both database and system administrators
  - Both groups have high damage potential

Client Based Attack Classification

5 - Administrative Users (cont.)

- Most Probable Attacks
  - Privilege abuse
  - Data destruction
  - Confidentiality breach (when administrative personnel should not be exposed to data)

- Require strong authentication on admin network connections
- Require individual identification
  - (i.e. NOT shared admin accounts)
- Audit
  - Mechanism should not be controlled by the admin
- Physical data redundancy
- Data encryption by app

Constructing an Effective Solution

Constructing an Effective Solution

Client Group Specification

- Identify and specify all client groups

- Example App Server Group
  - Source IP addresses:
    - 192.168.1.1, 192.168.1.2
  - Client software
    - (custom)
  - Database account
    - app_user1
  - Set of SQL Statements:
    - Select user_id from app_users where ….
    - Insert into orders values …
    - Select * from orders where...
  - Additional protocols
    - None

- Example Admin Server Group
  - Source IP addresses:
    - 192.168.10.1
  - Client software
    - None
  - Database account
    - None
  - Set of SQL statements
    - None
  - Additional protocols
    - SNMP, ICMP Ping

Constructing an Effective Solution

SQL Statement Identification

- How to identify and specify the set of SQL statements?
  - How would you identify the set of required database privileges?
- Solution #1: Ask Programmers
  - Never enough time to do it right
  - Things change over time
- Solution #2: Monitor Application Traffic
  - Solution must be capable of modeling SQL information

Implementing Countermeasures

- Derive a specification for countermeasures at all enforcement layers

	Audit	Statement Level ACLs	DB Connection Control	IPS / IDS	Network ACLs
App Server Group	🟧	🟩	🟩	🟧	🟨
Admin Server Group	🟧	🟩	🟩	🟧	🟨
Client Group 3	🟧	🟩	🟩	🟧	🟨

Implementing Countermeasures

- Network ACLs
  - Can use any network firewall
  - Optional: use VPN for secure administrative connections
- IPS / IDS
  - Should posses generic capabilities (i.e., Snort™ compliant)
  - Should have strong capabilities with regard to database network protocols

Implementing Countermeasures

Database Connection Control

- Decisions based on Source IP address, Target Account Client Software
- Enforced policies may include:
  - Access control (application specific accounts)
  - Failed logins
  - Number of concurrent connections
  - Time based controls (time of day, life span of a connection, etc.)

Implementing Countermeasures

Statement Level ACLs

- AKA: Enforcing the “set of known SQL Queries”
- Can be simplified by using object ACLs with wild-cards
- Delivers true control in real world scenarios:
  - 3 tier applications
  - Multiple applications using the same set of database tables

Implementing Countermeasures

Audit

- Accuracy / Granularity
  - It’s not enough to know that some application issued a “Select” statement against the credit cards table
  - We need to see the parameters of the query
- Independence
  - Mechanism should not affect the performance or behavior of the database server
  - Mechanism should not be modifiable by administrative users

Summary

- Classifying DB threats based on potential attacker groups allows us to design an efficient, effective model for protection

- Network and IPS / IDS layers are usually controlled by security officer
- Other layers are usually controlled by database administrators:
  - Implementation highly dependent on the specific database
  - Implementation is usually complex and costly
  - Security mechanisms rely on the secured object (paradox)

Summary (2)

- We need a new breed of security technologies that will put return control of database security to the security officer

- Such technologies should be
  - Independent of specific database platform
  - Independent of the database server itself

- Practically, the new technologies should implement all required protection layers within a single package

Q&A

Breaking the Rules: The Future of the Firewall

Shlomo Kramer, CEO and Co-Founder, Imperva, Inc.

Teale Data Center Technology Day

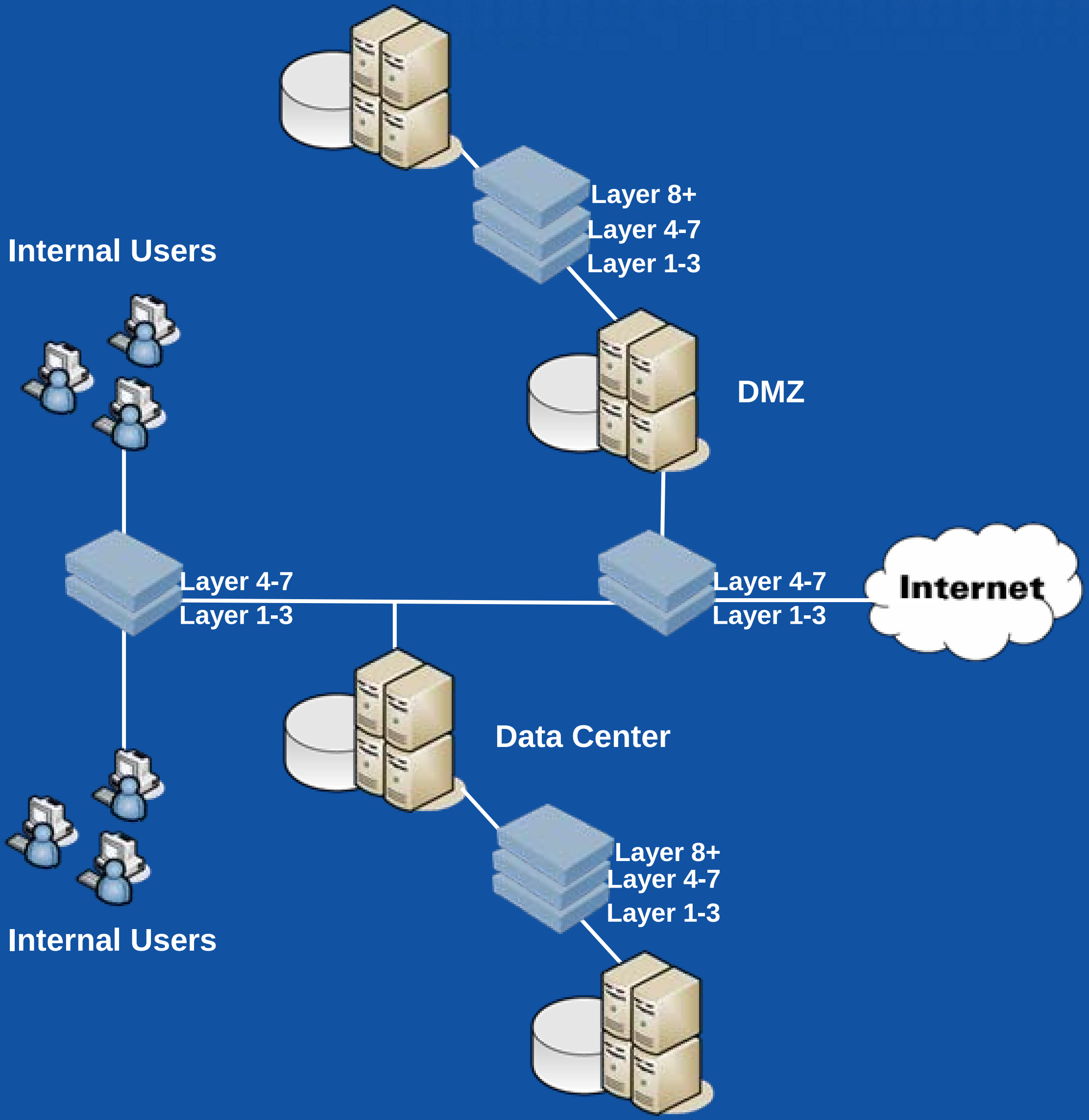
March 28, 2005

IMPERVA

Agenda

- Firewall Technology in the Network
- The Application Security Problem
- Assessing Current Approaches
- The Next Generation Firewall Technology
- Summary
- Introduction to Imperva and SecureSphere

Firewall Technology in the Network



1. Corporate networks connected to the Internet
   - Stateful Inspection deployed
     - Perimeter protection with simple DMZ
     - Some internal deployment
2. Worms became a nuisance inside the corporate network
   - Deep Inspection deployed
     - Internal deployment
     - Added to perimeter
3. Business applications get more sophisticated
   - New Technology required
     1. Inside the DMZ
     2. Inside the data center

Application Security Landscape

Need for Total Application Security

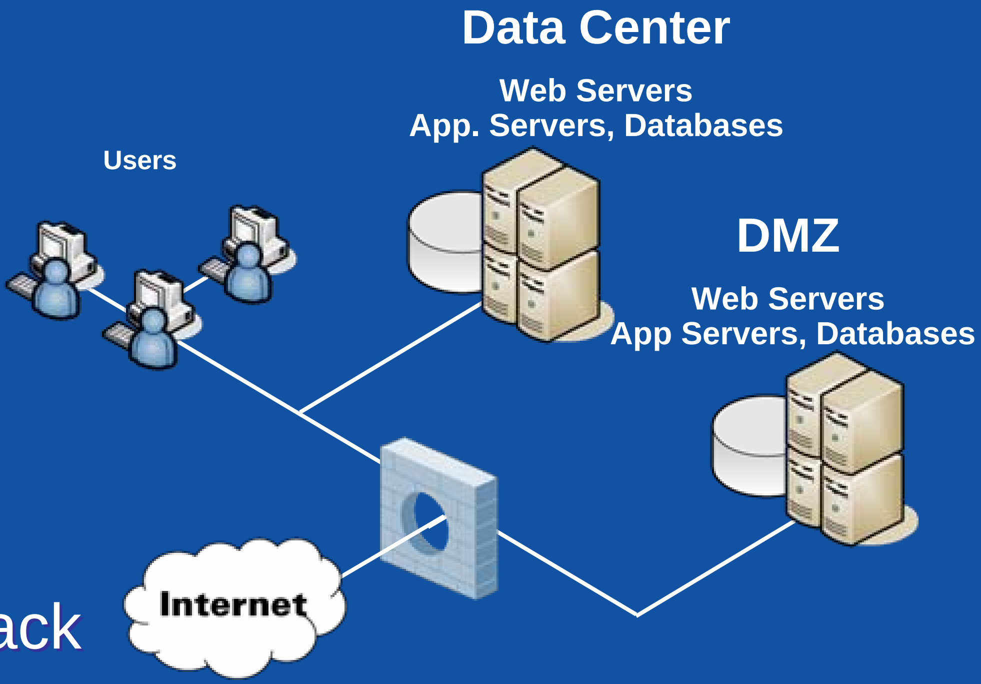
Applications have Never Been More Critical…

…or More Vulnerable

- **92%** of web applications are vulnerable to attack*
- **75%** of cyber attacks and Internet security violations are through Internet applications**

Business Implications of Attack

- Lost revenue
- Brand erosion
- Regulatory compliance issues



**Source: Imperva Application Defense Center*
*** Source: Gartner Group*

Application Security Landscape

FTC Punishes PetCo.com

- **Crime** - Vulnerable Customer Data

  Pet supply retailer PetCo disclosed this week that its security and privacy practices are the target of an investigation by the U.S. Federal Trade Commission (FTC), which is following up on an e-commerce security gaffe that left as many as 500,000 credit card numbers accessible from the Web earlier this year.

  - By *Kevin Poulsen*, Security Focus Dec 5 2003

- **Punishment** - 20-years "Hard" Audits

  Under the terms of the settlement, Petco is prohibited from misrepresenting the extent to which it protects the security of customers' personal information. The company must also establish and maintain a comprehensive information security program, certified by an independent professional every two years for the 20-year life of the order. Any violation can trigger an $11,000 fine.

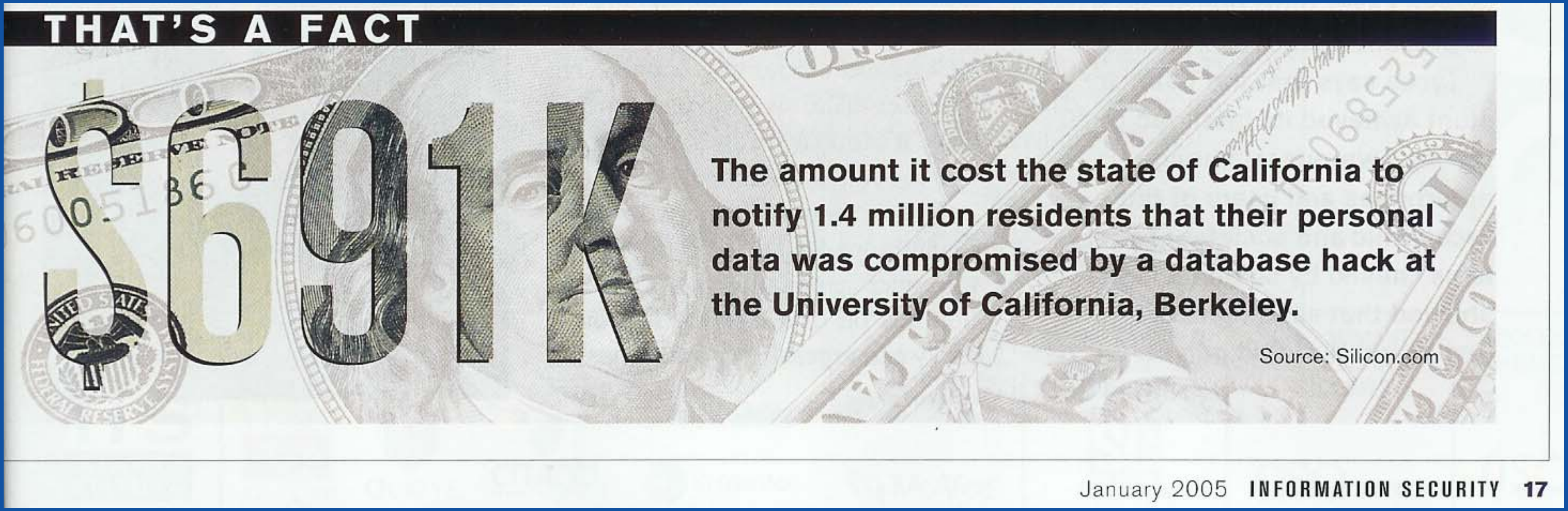
  - By *Kevin Poulsen*, Security Focus Nov 17 2004

Application Security Landscape

California Senate Bill 1386

- <u>Anyone</u> Doing Business with California Residents Must Notify Individuals of <u>Breach</u> Involving
  - Reasonable belief that
  - Unauthorized person acquired
  - Unencrypted personal information
- Personal Information Includes
  - Social Security Number
  - Drivers License
  - State Identification Card
  - Account, credit or debit card number
    - combined with security code, access code, or password to an access financial account.

THAT’S A FACT!



THAT’S A FACT

$691K

The amount it cost the state of California to notify 1.4 million residents that their personal data was compromised by a database hack at the University of California, Berkeley.

Source: Silicon.com

January 2005 INFORMATION SECURITY 17

What laws effect you?

CMS

Centers for Medicare & Medicaid Services

FISMA Implementation Project

Protecting the Nation's Critical Information Infrastructure

Sarbanes-Oxley

Public Company Accounting Reform and Investor Protection Act

ISO-17799

Information Security - As defined by ISO-17799, information security is characterized as the preservation of: Confidentiality - ensuring that information is accessible only to those authorized to have access. Integrity - safeguarding the accuracy and completeness of information and processing methods. Availability - ensuring that authorized users have access to information and associated assets when required.

Gramm-Leach-Bliley Act of 1999

HIPAA.ORG

Can Application Security be Solved in Source Code Alone?

- 93% of **retested** applications exhibited high-critical vulnerabilities

Vulnerabilities previously found, but weren’t addressed	14%
Vulnerabilities previously found and fixes were attempted unsuccessfully	16%
New vulnerabilities introduced between penetration tests	60%

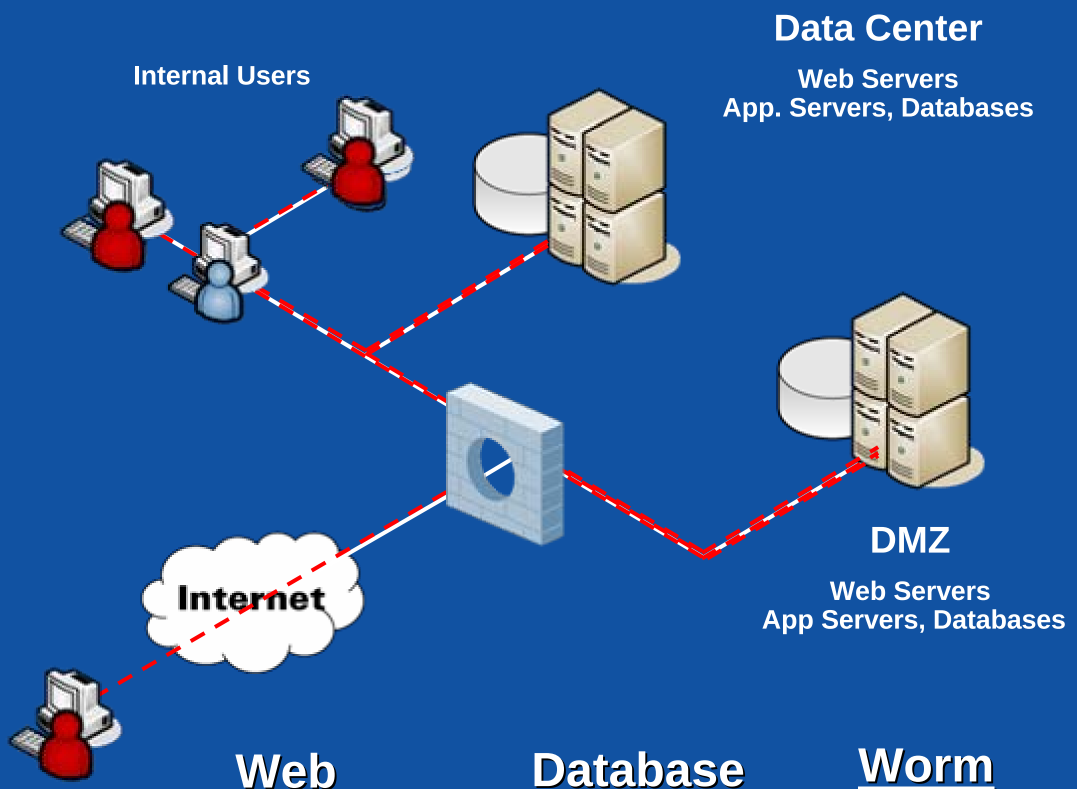
No – Additional measures are needed

**Source: Imperva Application Defense Center, 2004*

Application Threats

A multi-dimensional problem

- Web attacks
  - Targeted
  - External sources
  - “Custom” vulnerabilities
- Database breach
  - Internal sources
  - Very high value target
- Worm infection
  - External and internal sources
  - Generic attack



Web
SQL injection
Cookie poison
etc.

Database
Data theft
Data corruption
etc.

Worm
Code Red
Nimda
etc.

Security Coverage

Protect Entire Application Infrastructure in Data Center

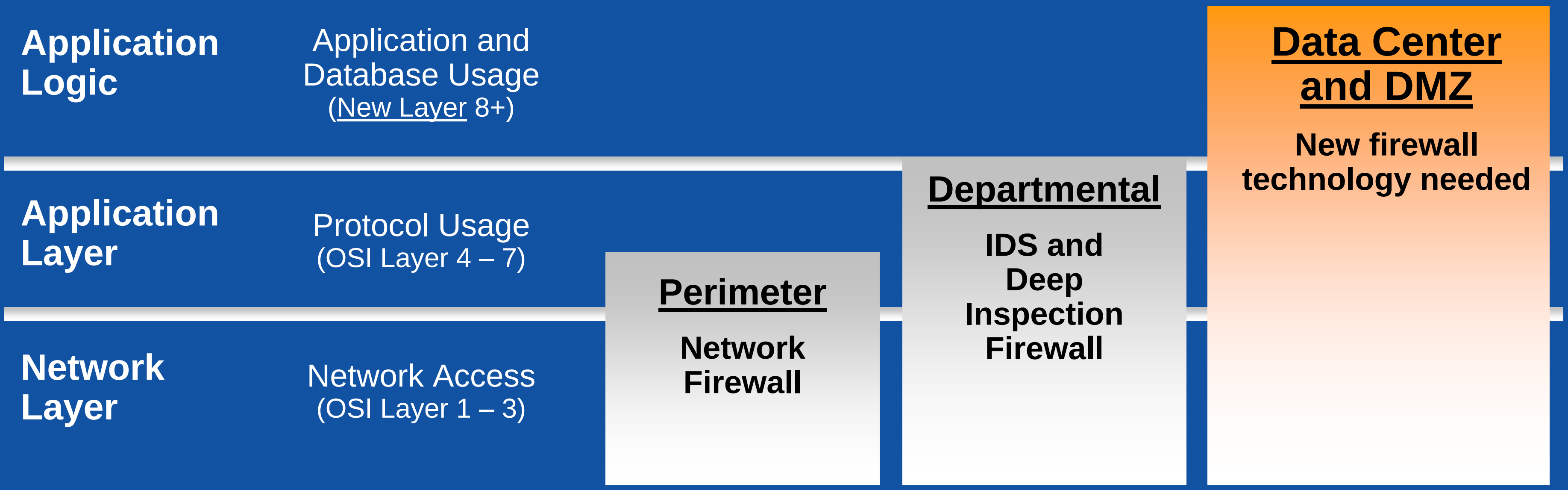
- Web Application (Requirement)
- Database (Requirement)
  - Via web application
  - Direct database attacks
- Worm Infection/ Platform Attack (Requirement)
  - Infrastructure Server Software
  - Network Stack
  - Operating Systems
- Web Services (Emerging)
  - Medium to long term issue for customers

Customer View of
“The Application”

Web Application or Web Service (Custom to Package)	Application Logic (Custom to Package)	Application Databases (Custom to Package)
Web Server	Application Server	Database Servers
Network Stack	Network Stack	Network Stack
Operating System	Operating System	Operating System

Total Application Security Requires
A New Type of Firewall

- Application Vulnerabilities are not Addressed by Existing Firewall or IDS Technology
  - SQL injection
  - Parameter tampering
  - Improper database access
- A New Type of Firewall is needed for App. Security
  - Complements existing network security investments



Application Logic

Application and Database Usage (New Layer 8+)

Application Layer

Protocol Usage (OSI Layer 4 – 7)

Network Layer

Network Access (OSI Layer 1 – 3)

Perimeter

Network Firewall

Departmental

IDS and Deep Inspection Firewall

Data Center and DMZ

New firewall technology needed

Firewall Technology Has Reached a Limit

Application Logic
New Layer(s)! 8+

Application Layer
(OSI 4-7)

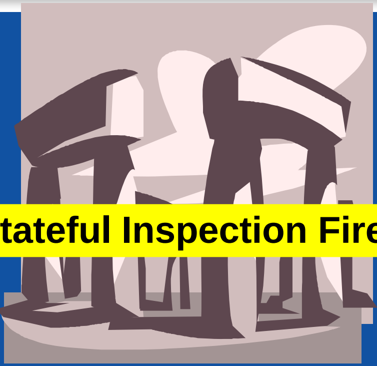
Attack Database
Hundreds of static rules

Predefined
Manually Tuned

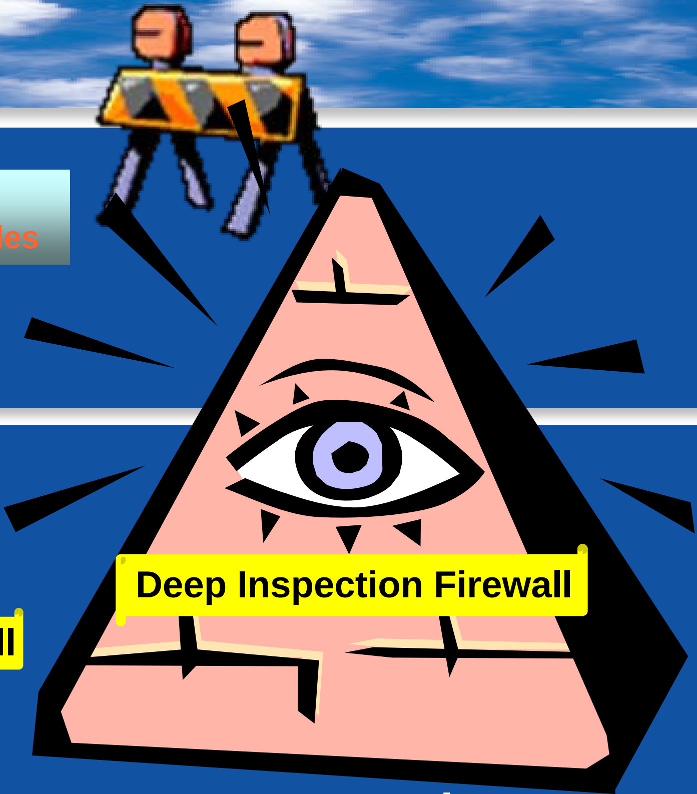
Network Layer
(OSI 1 – 3)

Rule Base
Dozens of static rules

Manually Built
Manually Tuned



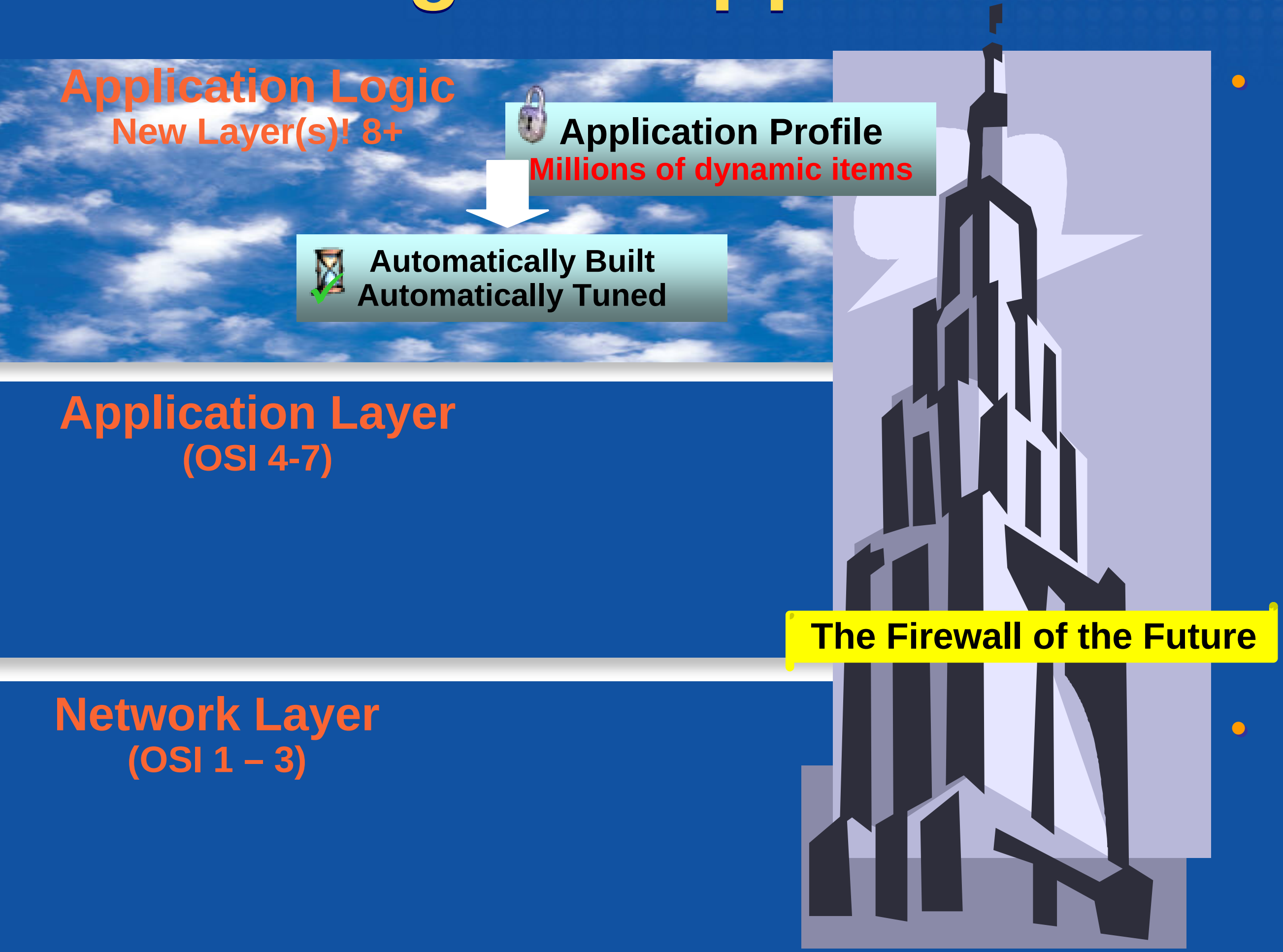
Stateful Inspection Firewall



Deep Inspection Firewall

Conventional Firewall Technologies Require Human Intervention to Tune Static Rule Bases.

Breaking the Application Security Barrier



- Much more information needed for security decisions
  - Application elements
    - URLs, Cookies, Parameters, Users, Sessions, etc.
  - Database elements
    - SQL Queries, SQL Tables, Users, etc.
  - Web Services elements
    - URLs, SOAP actions, XML files, XML fields & security constraints
- Too complex for manual intervention

An application security solution must build and tune the security profile without human intervention

Accuracy of Security

Application Security is not Black and White

Good

- Legitimate users and protocols
- Known good behaviors
  - Allowed URLs
  - Known parameters

Static “White List” / “Positive Security Model” blocks many legitimate uses

Suspicious

- New legitimate users or unauthorized users?
- Signature attack or free text input?
- Application change or reconnaissance?
- New / unknown good behaviors or multi-stage attacks?

Bad

- Unauthorized Users or protocols
- Known bad behaviors
  - Signature matches
  - Protocol violations

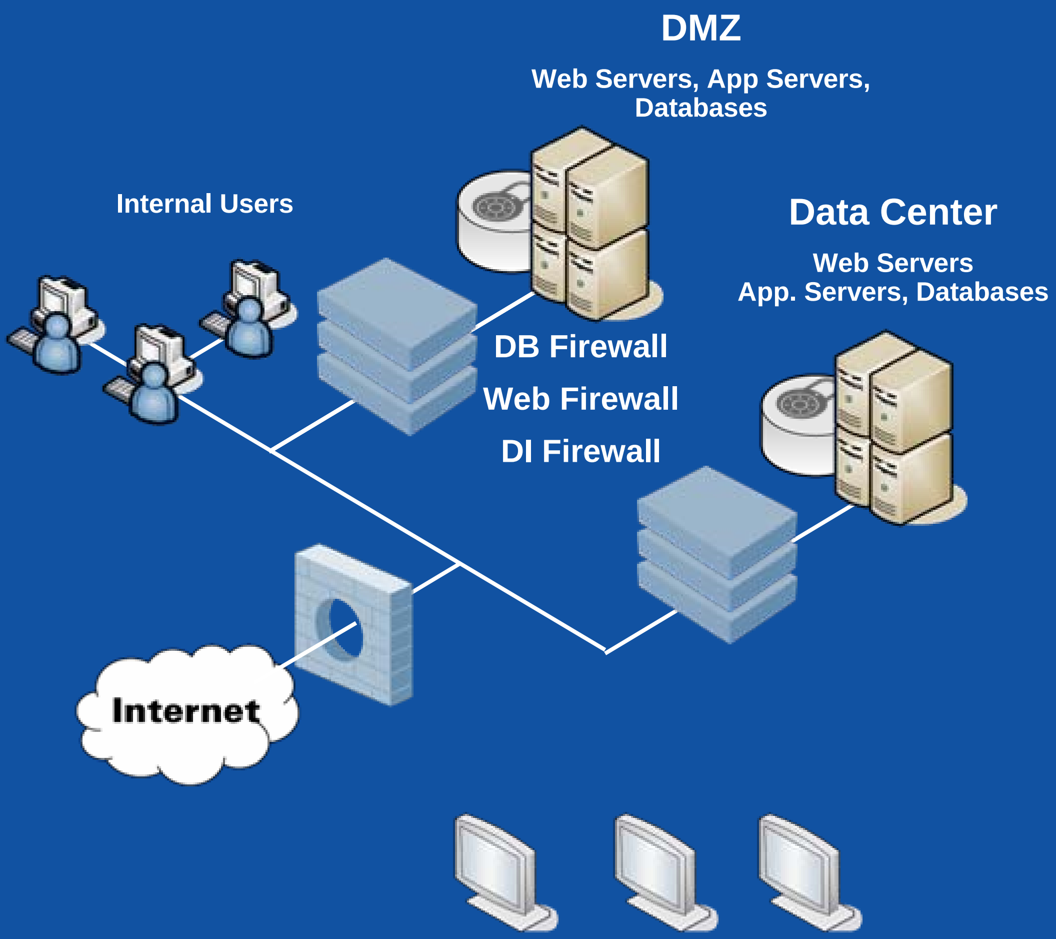
Static “Black List” / “Negative Security Model” allows many attacks through

Requirement: Application security solutions must address the complexity of “gray” traffic

Current Approaches

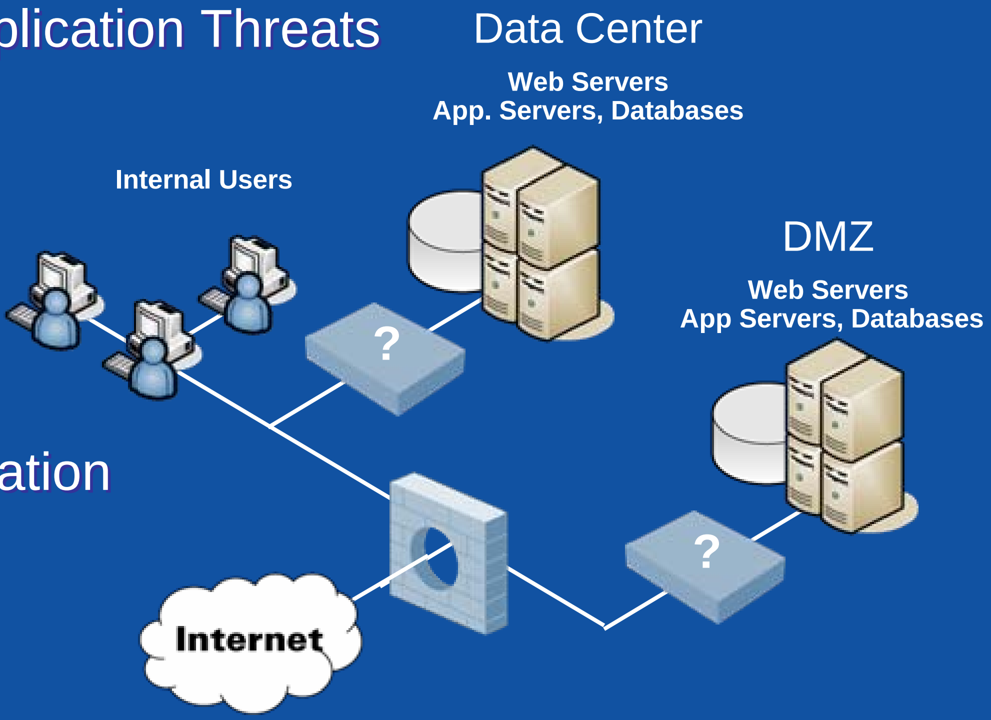
Comprehensive application security is a complex challenge

- Multiple products
  - Multiple interfaces
- Static policy/rules
  - Constant manual tuning
- Fragmented management
  - Set policy on each device
  - Fragmented logging, forensics, monitoring
  - No reporting
- Single points of failure



The Next Generation of Firewall

- Complete Protection From Application Threats
  - Web Attack
  - Database Breach
  - Worm Infection
  - Web Services Attack
  - Internal and external attack sources
- Automated Modeling of Application Structure and Dynamics
  - No on-going manual tuning
    - Must adapt as app changes
- Accuracy of Security
  - Capable of adapting as Application Changes
  - Capable of stopping "Under the Radar" attacks.
- Unified Management
  - Centralized policy, forensics, and reporting

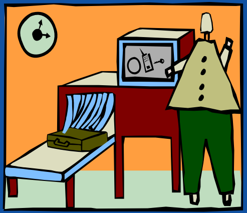


Matching Technologies to Security Challenges

Keeping the bad guys out; letting the good guys in. ✓

Stateful Inspection Firewall

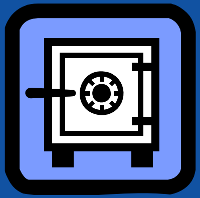
- Perimeter deployment



Preventing the internal spread of viruses and worms ✓

Deep Inspection Firewall + Anti-Virus

- Interdepartmental deployment



Protecting custom applications and data from targeted attacks ✓

Dynamic Profiling Firewall

- Data Center and DMZ deployment

Summary: Total Application Security

- Application vulnerability is a direct business threat
  - Lost revenue
  - Brand erosion
  - Regulatory compliance
- Application security requires a dedicated solution
  - Multiple threat vectors
  - Complex security requirements
  - Existing firewall technologies do not meet this threat
- Dynamic profiling technology is the answer
  - Full protection from Web, database, worm attacks
  - Automated learning
  - No manual tuning

Imperva’s Unique Heritage

- Expertise in Both Application and Network Security

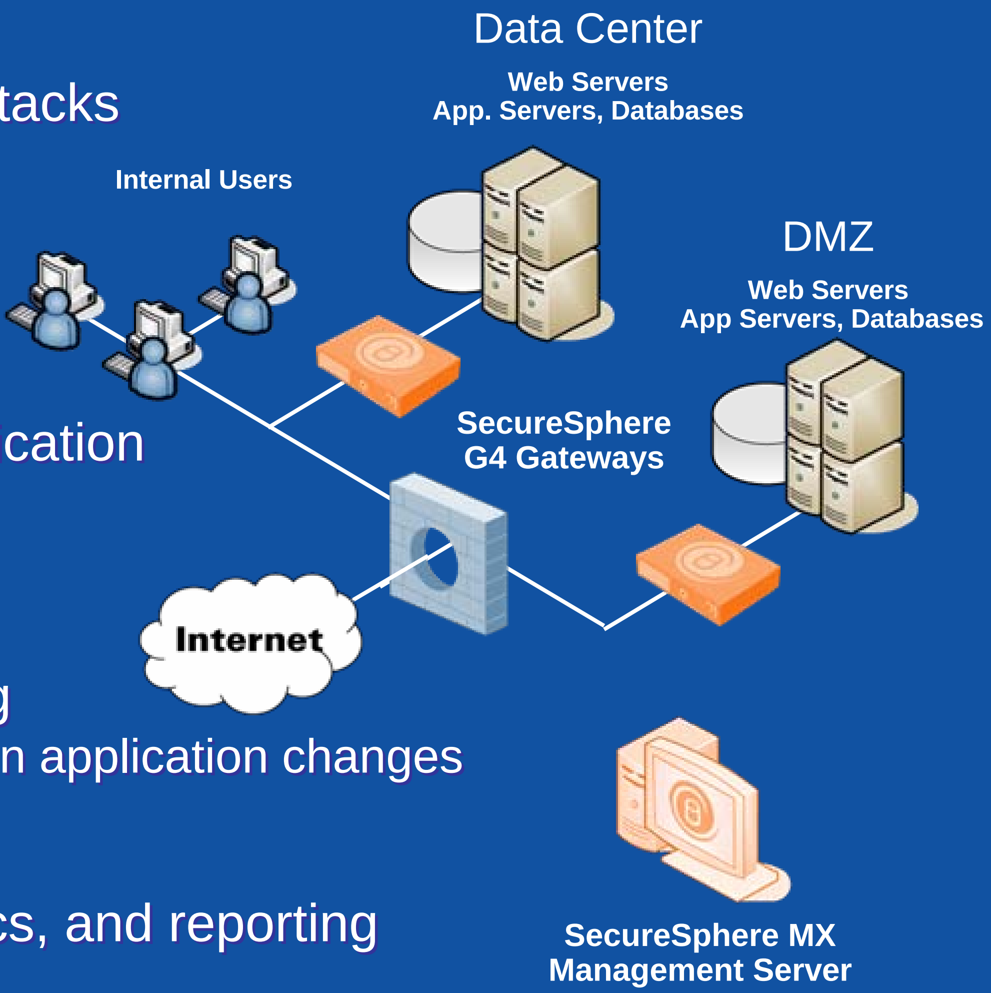
- Application Security - Elite Specialists
  - Israeli Defense Force cyber warfare team
  - Private sector application penetration testing and application security consultants

- Network Security - Shlomo Kramer
  - Check Point co-founder
  - Co-developer of Stateful Inspection
  - Imperva Co-founder and CEO

SecureSphere

The First Dynamic Profiling Firewall

- Unified Protection
  - Web, database or worm attacks
  - Internal and external
  - Layers 1-7 and 8+ (application and database)
- Dynamic Profiling
  - Automatically models application structure and dynamics
    - *Application*: URLs, cookies, users, parameters, sessions, etc.
    - *Database*: SQL queries, SQL tables, parameters, users, etc.
  - No on-going manual tuning
    - Automatically adapts when application changes
- Unified Management
  - Centralized policy, forensics, and reporting

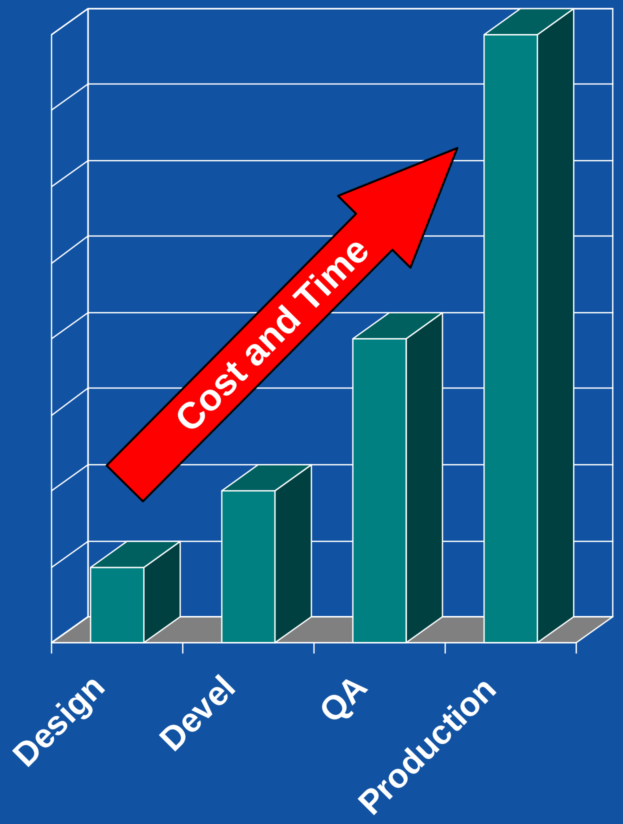


Application Security Life Cycle (ASLC) Integration

IT Operations ☞ App Development

- Cost and Time of Fixing Vulnerabilities Increases in Later Phases of Life Cycle
- Many Vulnerabilities Can Only be Found in Production Environment
  - Complexity of entire infrastructure
  - Identified by new attacks and probes
- Total Cost = Cost & Time to Fix + Risk + Lost Revenue
  - At risk until fix in production
  - Risk of “rush” fixes w/o sufficient testing
  - Roll-back of vulnerable functionality has significant business cost
  - No previous version to roll-back to

Cost and Time to Fix Vulnerability

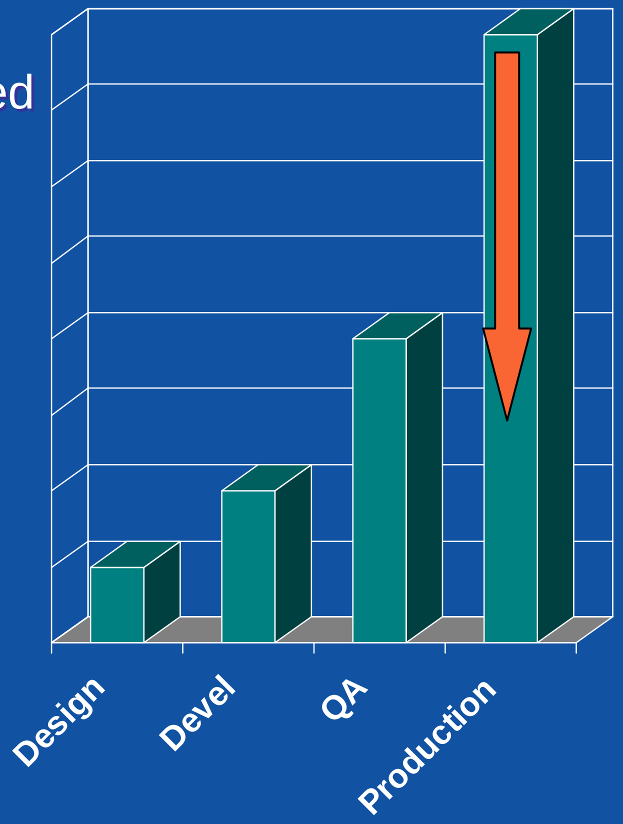


Application Security Life Cycle (ASLC) Integration

IT Operations ☞ App Development

- Immediate Protection
  - Web, database, platform, and network stack
  - Blocks attacks without manual configuration
  - Blocks unknown attacks
  - Blocks known attacks before production fixed
- Reduce Cost of Production Vulnerabilities
  - Fix vulnerability on *your schedule* not the attacker's schedule
  - Eliminate cost of "rush" fixes
  - No need to roll-back vulnerable release so no lost revenue
- No Changes Required
  - No application changes
  - No infrastructure changes

Imperva Reduces Cost to Fix Vulnerability



Questions?